METHOD FOR TEXT EDITING BY HAND DRAWN INPUTS

Reference to Related Application

This application is a continuation-in-part of U.S. patent application serial no. 09/880,397, filed June 12, 2001, which is a continuation-in-part of U.S. patent application serial no. 09/785,049, filed Feb. 15, 2001, for which priority is claimed. The entireties of the prior applications are incorporated herein by reference.

Background of the Invention

The present invention relates to methods and devices for editing text displayed on a screen and, more particularly, to pen input systems for editing text displayed on a screen.

The copending patent application captioned above describes a graphical user interface for a machine having a screen display. A significant feature of the interface is that it provides the means for hand drawn entry of on-screen objects which may be associated with functions, files, connections, and other objects or actions accessible by the system to carry out the user's desired purpose, whatever it may be. A salient aspect of this interface is the ability to carry out commands and actions corresponding to the hand drawn inputs as they are applied to any on-screen object, including (but not limited to) text presented as text files, ASCII files, or word processing files in standard word processing applications.

The hand draw approach utilizes hand drawn objects and various user-definable connects between those objects and various contexts of those objects (among other characteristics) to implement and control text editing functions. Furthermore, these hand drawing techniques increase the speed and familiarity and flexibility of a wide range of typical text editing processes, by replacing them with non-menu driven operations. Also, many of these prior art text editing operations can be utilized only according to set orders or set procedures. These orders and procedures, in many cases, can be eliminated by the present invention.

Using current technology, a computer user may employ a microphone connected to the input of a computer, to simply speak words intended to become text (dictate letters, documents, etc.). Improved voice recognition software and increased processing power have made this technology more commonplace today. As the text is spoken, it is processed by the software in the computer and presented (shown) on a computer screen as familiar text – in whatever font is the default for such a system. Generally, the software provides specific, limited techniques for entering punctuation in the spoken text. However, the most intuitive way to speak text is not to speak the punctuation, the paragraph start and stops and the indents, etc. Such text housekeeping "speak" is not easy and tends to interrupt the train of thought of the speaker. What is preferable is the ability to simply speak naturally (as if talking on the telephone) and then edit the text later.

Regarding editing after voice recognition input of text, it is well known that voice recognition software inevitably misunderstands some words, phrases,

and sentence structures. Although it is possible to direct the software to these errors and enter corrections by further voice inputs, this is a laborious task that slows the overall text input process and greatly impedes widespread acceptance of voice recognition software. Many voice recognition software users (and many non-users of computers) cannot operate a typewriter keyboard effectively, either because of disabilities or lack of manual keyboard skills. Clearly there is a need in the prior art for effective text editing that, like voice recognition inputs, does not rely on an alphanumeric keyboard and mouse, as do common word processing applications.

The hand drawing approach to text editing is such a system. It provides an arrangement to very quickly and intuitively edit text that may (after initially speaking or otherwise entering the text) be stored into a computer as one or more large paragraphs, i.e., many pages long. Hand draw editing enables users of spoken text to edit this text into a finished document very easily. This same technique can also be used for text typed on an alphanumeric keyboard, which may be many pages long and may need reordering, cutting and pasting, formatting, and/or other types of correction.

Currently cut and paste procedures have one thing in common: cut and paste or the copy and paste functions, as the two procedures are more commonly termed in word programs, are able to copy only one file to a clipboard and thereafter paste it as many times as desired. A problem often arises in dealing with a text document of some length, e.g., 25 pages, when it becomes necessary to move words, phrases, sentences and paragraphs from one part of the document to

determine if they work better in another part of the document. And one may wish to move multiple parts of multiple pages to multiple insert points on multiple pages. This is fairly common in word processing, but the current copy/paste approach is actually very inefficient for such tasks.

In fact, it is commonplace for computer users to make a "hard copy" (paper printout) of their document (such as the 25 page example) and then edit it with pen or pencil on paper. When it is necessary to reorganize and edit a long document, the hard copy edit seems for many to be the most intuitive way to explain to someone what type of reorganization is required. Thus the putative advantages of a computer word processing system are nullified, at least for this procedure.

To copy one section of text at a time and then scroll forward or backward in a text document and then paste that text to see if it works is tedious for many reasons: (1) one can only copy and paste one thing at a time (one word, one group of sentences), (2) one cannot in most systems copy one word of a sentence and then skip some words and then add to that "copy" some other words that are not contiguous -- you must copy contiguous text, (3) you must copy contiguous sentences and paragraphs, 4) whatever is copied is always text – it cannot be changed to another form, such as a functional object.

Another common problem with word processing editing involves the ability to easily navigate from one page of text, graphics, video, pictures and another. One common method of accomplishing this process is to "drag" text, objects, pictures, video clips, etc. from one page in a file to another. The problem

with this method is that the more pages you need to drag between pages or documents the longer it takes, because there is usually a "set" scrolling speed (onscreen) for "dragging" things. The other popular method for accomplishing this task is to cut and paste text, objects, graphics, video clips, etc.

An easier and more immediate way to navigate between multiple pages of a file or document is the use of Go To commands. Some programs do utilize Go To's, but they are cumbersome to use and involve finding the Go To function in a pull down menu and then activating it.

Currently, if one engages the heading software for some of the word processing applications, the software enters an "A" , for example, as the heading for a paragraph. When the user hits the return or enter key to drop the cursor down to a new paragraph, the software automatically enters a new heading letter. However, if one desires to type a second paragraph under the "A" heading, and hits the return key to start the second paragraph, the software automatically places the next outline heading "B". It is then necessary to delete this heading, move the tabs on a ruler to adjust the new text to line up under the existing "A" heading and type the second paragraph. Placing headings selectively in text should be more flexible and accommodating to the user's needs for multiple paragraphs under a single heading.

Summary of the Invention

A partial list of text editing functions made possible by the present invention:

1. Non contiguous (as well as contiguous) ability to edit text. A unique ability of this invention is the ease with which completely non contiguous words, phrases, sentences, paragraphs, etc., can be edited. Current text editing programs are largely, if not totally, limited to contiguous selection and editing of text. This invention provides an environment that is optimized for non-contiguous editing of text. Below is a partial list of non contiguous editing features (which can also be contiguous, if desired):
    - Delete
    - Copy
    - Paste
    - Change case, font, style, etc.
    - Indent
    - Make bold, italic, underline, etc.
    - Reorder
    - Go To
2. Many different types of text can be selected and copied and assigned or copied to a drawn object. This single object can contain as many selections of copied text as desired and each of these copied selections of text can be accessed at any time.

3. Drawn objects, which have one or more selections of text assigned, copied, etc. to them, can in turn be drawn anywhere in the context of text. Then the drawing of these objects can be used to insert the text that is contained within them into the text in which these objects are drawn.
4. Multiple choices for text edits can be assigned to multiple different drawn objects, and these drawn objects can be used to select entire sections or complete documents that have been edited a certain way. By using multiple hand drawn objects and assigning multiple different versions of edited text to them, one can touch each drawn object to see the edited version contained therein. By this method one can quickly compare multiple edit versions for a single text document or its equivalent.
5. The user has constant access to the unaltered Original (Raw) text to compare to the edited version of the same text. The original text can be maintained with its original edit marks, objects, and their equivalents, while the results of these edits can be viewed as separate text (in a separate window or as separate text) that is updated with each new edit.
6. The ability exists to instantly "travel" between individual edited text, as shown in separate text like in a Print Preview Window showing the "Edited Text Result", and places in the Original (Raw) text where edit markers or objects or their equivalent have been drawn, inserted, dragged to, or otherwise placed in the original text to indicate an edit. In other words, users of this invention can travel between the Edited Text Result and any individual edit marker, object or its equivalent as placed in the

Original (Raw) text and vice versa. This provides an environment that is optimized for permitting the constant changing of one's mind in the editing process.

7. Colors can be used to differentiate between drawn objects of the same type, i.e., stars, check marks, rectangles, etc.
8. Colors can be used to designate actions, functions and the like that have been assigned to lines or arrows. Colors can equal "edit logics".
9. Colors can be used to distinguish one group of selected text or text objects from another.
10. Text can be converted to graphic objects, which in turn, can be used to store commands or other text, music, graphics, video, etc. Text objects can be directly used to effect the processing of audio and video and virtually any type of computer data.
11. Multiple different words can be selected at once and only these words and, if desired, only selected instances of these words, can be spell checked at one time.
12. This invention obviates the need for a myriad of pull down menus to access text edit commands and functions and replaces these menus with a selection of user-defined drawn objects. These objects can be used for storing, selecting, and designating, etc,. any text and for applying edit functions, actions, applications and the like to any text.
13. Conventional Go Tos can be replaced with user-definable drawn objects, which can be drawn anywhere in text, menus, pictures, etc.

14. Graphics can be drawn or engaged to invoke voice recognition as an editing tool, plus drawn graphics can be used to select text which is to be verbally edited.

15. Drawn graphics can be used to define outlines and apply these outlines to text. Furthermore, drawn graphics can be used to select or define a portion of an outline, (including heading, subheads and sub-subheads) as a single graphic object. This graphic object can then be drawn, dragged, copied, etc,. to insert this outline section in an existing outline. The advantages here are ease of use and the fact that as a graphic object the structure of the (copied) portion of any outline assigned to a drawn object, remains intact; that is, the indents do not disappear, the subheads and sub-subheads do not change their numbering schemes, etc.

16. Functions of drawn objects, i.e., the order of selected text, can be automatically sequenced, permitting text lists to be easily reordered or have their grammar changed, etc.

Brief Description of the Drawing

Figure 1 is a layout showing raw text with hand drawn inputs to reorder the sentence order, and the resulting edited text.

Figure 2 is a layout showing raw text with hand drawn inputs to place text in onscreen hand drawn objects, and the resulting edited text.

Figure 3 is a layout showing raw text with hand drawn inputs to delete text portions, and the resulting edited text.

Figure 4 is a layout showing raw text with hand drawn inputs to place multiple text selections in an onscreen hand drawn object, and the resulting edited text.

Figure 5 is a layout showing raw text with hand drawn inputs to reorder the sentence order, and the resulting edited text.

Figure 6 is a layout showing raw text with hand drawn numerical inputs to reorder the sentence order, and the resulting edited text.

Figure 7A-7F are layouts showing various techniques for selecting a text portion and placing it into an onscreen hand drawn object.

Figure 8 is a layout showing raw text with various hand drawn inputs that may be used to insert text or other material in a text portion.

Figure 9 is a layout showing raw text with hand drawn inputs to select text portions and change them to bold, and the resulting edited text.

Figure 10 is a layout showing one technique of the present invention for entering a "Go To" command.

Figures 11A and 11 B are layouts showing raw text with hand drawn inputs to reorder the words in a list format, and the resulting edited text.

Figure 12 is a layout showing the word list of Figure 11 with hand drawn inputs to reformat the list as an outline, and the resulting edited text.

Figures 13A and 113B are layouts showing the outline of Figure 12 with hand drawn inputs to change and correct the punctuation , and reorder the outline, and the resulting edited text.

Figure 14 is a depiction of a typical Info Window for an arrow transaction such as punctuation correction and spelling correction.

Figure 15 is a layout showing raw text with hand drawn inputs to select text portions and change them by means of voice recognition input.

Figure 16 is a layout showing raw text with hand drawn inputs to select words and change to italic, check spelling, and delete text.

Figure 17 is a layout showing raw text with hand drawn inputs to reformat the text into an outline/heading format.

Figure 18 is a layout showing raw text with hand drawn inputs to reformat the text into an outline/heading format.

Figures 19A-19C are a sequence of views showing a method using hand drawn inputs for reformatting a text list into a plurality of separate graphic objects which represent one or more elements existing as readable computer files, such as .WAV files, JPEG files, video files, data files, and the like.

Figures 20A-20B are a sequence of views showing a method using hand drawn inputs for sorting items from a text list.

Figure 21 is a layout showing an original email message edited by drawn inputs in accordance with the system of the present invention.

Figure 22 is a layout showing a list of graphic objects existing as individual words, and a technique for selecting some of the objects by color and applying an action to modify a characteristic of each graphic object's contents.

5

Description of the Preferred Embodiment

The present invention generally comprises a system for editing text in a computer or word processor using hand drawn inputs. The invention is based on the use of hand drawn onscreen objects, and the use of arrow logics, as described in U.S. patent application serial no. 09/880,397, filed June 12, 2001, and U.S. patent application serial no. 09/785,049, filed Feb. 15, 2001, referenced above and incorporated herein by reference. The text to be edited may be acquired by voice recognition software input, OCR input, keyboard entry, use of previously entered text stored on hard drive, floppy drive, CD-ROM, or similar media. Likewise, the text may be acquired by transmission via email, email attachment, FTP link, or the like.

In the description below, the term "arrow" can also indicate a line that is programmable to convey a transaction. The transaction can be an action, a function, a link, a cause and effect, an association, etc., between two or more objects.

With regard to Figure 1, both the Original (Raw) text and the Edited Text Result are shown as two separate pages or windows. Users can travel from one window to another by clicking on edit objects or marks on the Raw Text which enables them to instantly go to where this text has been inserted or deleted from the Edited Text Result window. Likewise, users can click on colored text or small drawn objects present in this text to travel back to the place in the Raw text where the original edit command object was drawn or placed.

A block of text is displayed for editing by the user, the text comprising quotations from Mark Twain, the great American author. The user may wish to select and reorder the sentences to change the meaning, or rhythm, or sequence of ideas of the presentation. In accordance with the invention, one method for selecting a sentence is to hand draw one or more open or closed curved loops about the sentence, including all the words thereof. Thus, for example, the sentence "In the center ...are you young." is selected by drawing a loop in magenta line that encompasses the sentence. Likewise, two loops are drawn in dark blue to encompass all of the sentence "Twenty years ... you did do." to select that sentence, and a red line is drawn about the sentence "The miracle ...determined spirit." to select it. Other methods for selecting sentences are described below.

The goal being to reorder sentences in this example, the user may draw an arrow from one of the selected sentences to a place in the text where the user wishes to move the sentence. Thus, an arrow may be drawn from the dark blue loop surrounding "Twenty years ... you did so" to the beginning of the text block at the upper leftmost position thereof. The arrow logic (i.e., the transaction that is commanded by the arrow) is set by the color of the arrow, which is selected by the user in accordance with the transaction the user wants to carry out. The user may employ an Info Window (see Figure 14 for example) to set forth the relationship between arrow color and the transaction. In this example of Figure 1, the user has designated a red arrow to direct that the selected text at the tail of the arrow is deleted from its original position and inserted in the text at the

head of the arrow. The software interprets this arrow logic command, and carries it out in the edited text result, by changing the font color of the sentence to the color of the selection loop and moving the sentence to the beginning of the text block. Note: Info Window in this disclosure is defined as any menu, window, dialogue box, text group, or the like that is used to define or program any parameter.

Another technique for moving sentences involves entering the ordinal number of a selected sentence by writing the number adjacent to the sentence in a line having the same color as the selection loop. Thus, in Figure 1 the number "1" is written in red line adjacent to the red loop that selects the sentence "The miracle ... determined spirit." to command that the selected sentence be placed first among the sentences of the text block. However, if there is a command that arrow logic movements supercede ordinal number commands, then the selected sentence "The miracle ... determined spirit." is placed directly after the sentence "Twenty years ... you did so". Note that the color of the hand entered "1" clearly associates it with the sentence selected by the same color line, and avoids confusing associations with nearby sentences selected by other colored lines. Likewise the sentence "In the center ...are you young." may be labeled with a "2" in magenta line to place that sentence after the sentence "Twenty years ... you did so.". In each case the sentence that is selected is changed to a font color that corresponds to the respective onscreen selection loop color, and moved in accordance with the hand drawn editing commands, as shown in the Edited Text Result.

It is important to note that a display of the raw text with hand drawn editing inputs may be maintained onscreen, and a Print Preview window (or its equivalent) may display the Edited Text Result as shown. Thus the user retains the original text and the drawn editing inputs drawn thereon, making it far easier to compare and evaluate the changes and further alter these changes that the user enters. Many different changes in style and syntax and formatting may be entered, and the results weighed, without losing the original text. This feature encourages the user to try many different forms and edits, since the original text remains available if corrections or changes do not work well, and must be removed or redone.

Moreover, the color of the words or sentences in the Edited Text Result clarify which text portions have been changed from the original Raw Text. If the user is not satisfied with the edited result, the user may simply touch the pen or stylus (or mouse cursor) to any colored text portion, in order to be returned to the corresponding portion of the Raw Text version with its hand drawn inputs, whereby the user may alter or remove the hand drawn inputs to revise and re-edit the text.

With regard to Figure 2, another editing function that may be accomplished with the invention involves selecting and placing text within an onscreen object that is drawn by the user. The user may draw a rectangle (in blue line, for example) adjacent to the text block, and then draw one or more closed (or partially closed) loops, let's say in red, that includes all the words of the sentence "The miracle...determined spirit.". The arrow logic for this red arrow (which

includes the partial ellipse drawn around the text to select it) may be : "Send the object(s) that the arrow is drawn from to the object that the arrow is pointing to". In the case the "objects(s) that the arrow is drawn from equal selected text. Note: this arrow logic can be set up by a user in a menu.

This same red line may be used to select the sentence "In the center ... are you young.", and a red arrow extended to a yellow star object drawn on screen. This latter sentence is similarly placed in the yellow star object. The selected sentences may be removed from the Original (raw) Text and placed in the respective onscreen objects (or they could be copied and placed into these onscreen objects without deleting this text from the original text). This is dependent upon the arrow logic for the color red. In the case of Fig. 2, the arrow logic for the red arrow deletes the selected text from the Original (Raw) Text when it places that text in either the yellow star or blue rectangle.

The results of these editing inputs is shown in the Edited Text Result of Figure 2, in which the remaining sentences of the Raw Text are displayed in their original order, the two selected sentences have been removed, and the two objects, blue rectangle and green star, are displayed in miniature at their respective locations in the Edited Text Result. The user may touch either of the miniature symbols to cause the display to return to the Raw Text display and in particular to the portion of the Raw Text at which the symbol was drawn. Thus further editing may be carried out if desired. Alternatively, the miniature symbols may be dragged or copied and dragged, or just drawn at any point in the Edited Text Result to place the contents of each symbol at any other selected point in

the text. Thus sentences (and other text portions) may be deleted, and yet remain available onscreen for insertion at any other point in the text, or in some other text block, or stored for undefined further use, or changed. Given that arrow logics are user-specified, an alternative arrow logic may be defined so that the selected sentences are copied into their respective onscreen objects, without being deleted from the text.

It should be noted that the contents of any onscreen object may be utilized at any time and at any location in the text, in accordance with the following example: a (blue) rectangle (or any other object) may be drawn on or adjacent to any text block or portion, and an arrow may be drawn from the (blue) rectangle to a selected insertion point in the text to cause the contents of the (blue) rectangle to be placed at the insertion point.

In the descriptions in all the examples herein, it is important to note that the color of an arrow is selected to define the transaction that it carries out, while the color of an onscreen object (rectangle, star, folder, etc.) is used to define its identity and to distinguish it from other onscreen objects of the same shape but of a different color.

With regard to Figure 3, another simple method for removing sentences from a Raw Text block involves drawing an X figure over the sentence to be removed. The software recognizes the X, determines the sentence or sentences over which the X is drawn, and deletes those sentences from the Raw Text. The result, as shown in the Edited Text Result, is that the sentence "Whether sixty...of living." is deleted from the text. Another simple graphic gesture to

command a text delete function is a scribble that overlays a substantial portion of a sentence (the portion threshold may be user defined to accommodate the user's writing style). In Figure 3, the sentence "The miracle...determined spirit." is selected and deleted by a scribble mark drawn by hand thereon, and the Edited Text Result shows the consequential deletion of that sentence from the text.

Note that the Edited Text Result of Figure 3 also displays a miniature symbol depiction of the hand drawn entry that has been used to delete a portion of the text. The X or scribble is displayed at the point in the Edited Text where the respective text portion has been deleted. The user may touch (via pen, stylus, cursor click, etc.) the miniature symbol to command that the Raw Text display return to the text portion from which the deletion was taken, whereby further changes may be made if necessary.

As shown in Figure 4, a selection line may be drawn in an open or closed loop about one sentence (Age is ... matter.), another selection line may be drawn in a loop about another sentence (Twenty years ... you did so.), and a line or arrow may be drawn to connect the selected sentences in a desired order. The selection lines are drawn in a color picked to distinguish the selected text from other selected text. In other words, blue lines drawn around various portions of text to select it, distinguish this selected text from other text that was selected by drawing a green line around it., and this is distinguished from text selected by a pink line and so on. So the color of a line used to draw a selection loop or an object is used to distinguish the selected objects (in this case text) from other

objects of the same kind, or hand drawn objects, like stars, check marks, etc., from other objects of the same kind.

Note: The techniques for selecting text described herein enable the user to select any number of disparate text portions without having to make a commitment to copy and paste functions. The selected text can be edited or assigned (copied) to a drawn object, which in turn can be used to insert this text at a new location or be used to alter the text assigned to the object at any time and in any order subsequent to the selection process.

There are two general ways of selecting text: (1) drawing a graphic of some type around, under, over, next to, etc., some text that is desired to be selected, and (2) drawing a line around or partially around text and then continuing the same line that formed the loop to draw an arrow which points to some graphic object or text object or its equivalent. According to #2, the process of selecting and assigning, editing, copying, etc., is done with one stroke. According to #1, the process of selecting is done with one or more strokes and the process of assigning, editing, copying, etc., is accomplished with another stroke (which is a line or arrow that has a defined "arrow logic" assigned to it). This additional stroke with its arrow logic can be an action or it can be a function that is created or it can be an association or relationship between two or more objects, etc., that is carried out when this line or arrow is drawn. Note: another benefit of this invention is the ability to enable the process as described under #1 above to be used to create a memorized order of events, which are later used with regards to an edit function. In other words, the order that things (text) are

selected can be memorized as part of the selection process and then when an arrow logic is invoked for this selected text, the edit function of this arrow/line logic will be applied to this text in the order that is was selected.

In the case of Figure 4, the two sections of selected text could have been selected according to the #1 or #2 approach above. Either way, the result can be the same. However, in the case of #1 above, the color of the select line would not necessarily equal the color of the "arrow logic" ("edit logic") line. Any color can be used to select text or objects. Again, color for this purpose is used to distinguish one object of the same type, i.e., a star, from another object of the same type; or it is used to distinguish one group of selected text (i.e., a blue group) from another group of selected text (i.e., a gray group). Conversely, the color used to draw a line or arrow that conveys an "arrow logic" is designed to denote a specific type of function, action, association, link, etc., that is to be applied to the selected object(s) (which can be text).

One of the advantages of using color to select objects only, is that a user can create numerous color groups of selected text before doing anything with this text. So in this case, using color for selection permits users of this invention to pre-select multiple groups of text (i.e., text that they know needs to go together) before they do anything with this text. Then later, one or more colors of lines can be utilized to draw lines or arrows that convey certain user-programmed arrow logics ("edit logics") to any one or more of these selected groups to perform edits with them. This entire process serves to enable a much more intuitive editing process as commonly practiced "copy then paste" or "cut

then paste" is eliminated and replaced with the ability to select only. Then later copy or paste or cut any of this selected text at will and in a user's own time.

Referring again to Figure 4, the line/arrow color (red) is selected to command the transaction "group and place the selected text portions in the order in which the connecting line joins the text portions." At a later time, the user may draw a symbol such as the inverted triangle in blue. Then a colored line (in this case red) that represents an arrow logic (edit logic) is drawn from the selected text to an object (a blue inverted triangle). The drawing of this red arrow places the selected text (all of the selected text that is blue) into the blue inverted triangle. Note: the inverted triangle could have been any color. The color of the object (i.e., the inverted triangle) is not in this case a determinant of the types of colors of selected text that can be assigned to it. The color of the object is merely a way of distinguishing it from other objects of the same type.

At the point illustrated in Figure 4, the inverted triangle contains the text that has been selected in the paragraph above it. Thereafter, whenever the inverted blue triangle is drawn again onscreen, it represent and contains the selected sentences in the directed order. If the Object Info Window, or its equivalent, for the inverted triangle is invoked..... (continue with existing text). If the Object Info Window for the inverted triangle is invoked, it will display the two sentences in the set order, as shown in Figure 4. If the same triangle is drawn in the same color within a text block, the action may be set so that the text contained in the triangle is inserted at the place marked by the depending point of the triangle.

In general, the text portion that is placed in a hand drawn onscreen object, or which is deleted by a hand drawn deletion figure (such as the X or scribble described above) remains associated with the hand drawn input. The Info Window for any object may be displayed to visualize the text content of the object, so that the content is not lost, even in a deletion operation.

With regard to Figure 5, the invention also enables a user to select and reorder sentences by drawing a line from sentence to sentence. Each sentence may be selected by placing a loop of the line over at least some portion of the sentence. Thus, a line drawn over the Raw Text that begins with a loop over the word "lure" selects the sentence "Whether sixty...game of living.". The line continues therefrom to loop over the words "over matter", thus selecting the sentence "Age is...over matter." and placing it second in the order of selection. The line then continues and loops adjacent to the word "don't", thus selecting the third sentence as "If you...doesn't matter.", and so on. The reordered sentences appear as shown in the Edited Text Result of Figure 5.

The hand drawn editing inputs that are applied to any text portion may be saved for further use. For example, as shown in Figure 5, the drawn editing inputs drawn to change the order of the sentences may be stored in a blue folder (defined by an Info Window for this use) by drawing a (typically) red arrow from the text block to the folder. If the logic for a red arrow is "Place the objects that the arrow is drawn from into the object the arrow is pointing to", this text will be placed into the blue folder. Thereafter all the drawn editing inputs currently displayed disappear from the screen and are stored in the blue folder, but may be

recalled to the Raw Text and reapplied thereto. Alternatively, the symbol may be drawn in the Edited Text Result and to cause the reordered sentences to be displayed as shown in Figure 5. Note: This described operation could be modified to select only the words that the loops intersect. This would be a user selectable option: (a) select the entire sentence, or (b) select only the intersected words.

A further technique for reordering the sentences of a Raw Text block, as shown in Figure 6, involves the user writing the ordinal number of a sentence over a portion of the respective sentence. In this example, the numbers 1-12 are written over selected sentences, and thereby selected and placed in the desired order, as shown in the Edited Text Result of Figure 6. In addition, a blue star (or any selected symbol and color combination) may be drawn adjacent to the raw text, and a (typically) red arrow is drawn from at least one of the hand drawn numbers to the blue star. The blue star thus contains the drawn editing inputs of Figure 6. The user may desire to assay the differences in the two edits (Figures 5 and 6) of the same text and determine the better one. This may be accomplished by drawing either symbol in the Edited Text Result to command the display of the text contained in that symbol. Any set of drawn editing inputs and their resulting text may be saved and recalled (by redrawing the object that contains them), whereby many different edits may be tried. As a condition, the original text is not altered, but the edited text result display is immediately updated to portray the editing inputs currently onscreen.

The invention further provides several different methods for selecting text as a paragraph before manipulating it, such as directing it into an onscreen object. As shown in Figure 7A, the user may draw an arrow extending through all the lines of the text to be selected, with the arrowhead pointing to and in proximity to an onscreen object, such as the inverted triangle shown. Likewise, an arrow drawn as a loop extending over the lines of the text to be selected (Figure 7B) and pointing to an onscreen object accomplishes the same task: selecting the lines of text and placing them in the object at the head of the arrow. With regard to Figure 7C, the user may draw a single bracket extending vertically through the lines of text to be selected, and then draw an arrow from the bracket to the onscreen object to place all the text lines enclosed by the bracket in the object. A slight variant of this technique is shown in Figure 7D, in which the user draws a horizontal bracket above a paragraph to select the entire paragraph, and then draws an arrow from the bracket to the onscreen object to place the paragraph in the object.

Another technique for text selection, shown in Figure 7E, involves drawing a selection line that circumscribes a substantial portion (set by a user defined threshold) of a paragraph. An arrow drawn from the paragraph to an onscreen paragraph places the paragraph in the object. With regard to Figure 7E, a conventional selection rectangle known in the prior art may be used to select lines of text, and an arrow extending from the selected lines to an onscreen object places the lines in the object. It may be appreciated that Figures 7 show exemplary techniques for selection of text, based on the freedom of hand drawing

and the flexibility of user-defined functions and contexts for onscreen objects and arrows, and that a far more extensive palette of objects and arrows may be created easily by the user to suit the particular tasks being undertaken. Any of the selection techniques may be used instead of the open or closed selection loops described elsewhere in this application.

Note: If a blue arrow is used as shown in Figure 7a – 7F and this arrow places selected text into an object, then the logic for this arrow would be something like "take the selected objects that the arrow is drawn from and place them inside the object the arrow is pointing to".

Figure 8 depicts several different possible onscreen objects, and their use in inserting text (or other editing commands) in a block of text. One object is a loop that contains some text or one or more edit commands, the loop having a tail extending to the precise point in the text block where loop-contained text is to be inserted or the edit command is to be applied. Another object is a check (tic) mark, which contains some text or one or more edit commands, the vertex of the check mark indicating the point at which the text is inserted or the edit command is carried out. Likewise, an astrological symbol (Venus) and an inverted triangle (shown previously) may be drawn in the text block, the lower apex each indicating the insertion point. Any of these objects may be drawn in any one of a number of colors, each color defining a discrete object among the same shapes. Given a color palette of 30 colors or more, there is ample opportunity to provide the user with a large arsenal of distinct objects for a wide variety of editing

operations. Likewise, objects other than those shown in Figure 8 may be defined and used for editing, based on the personal preferences of the user.

The invention further enables formatting changes to be entered easily and naturally in text. As shown in Figure 9, the user may underline words to be selected using blue lines in various separate portions of the Raw Text, whereby all text thus selected is grouped for any subsequent transaction. Subsequently, the user may write a "b" adjacent to the text, and an arrow extending from the "b" to at least one of the blue underlined portions. This hand drawn input is interpreted by the software to command that all of the disparate words, phrases, etc. selected by blue line are placed in a bold font, which is shown in the Edited Text Result. This is an example of non-contiguous editing with this invention. A user could underline a group of words, letters, phrases and/or sentences all through a multi page document and then by drawing a single "b" and an arrow to any blue line, all text underline in blue becomes bold, wherever it is in the document. The changed text may also be depicted in the color of the inputs which created the bold change, herein shown as blue. As described previously, the user may touch (via pen, stylus, or cursor click) the blue text in the Edited Text Result to command the display to return to the respective portion of the Raw Text.

Likewise, the user may draw a green arrow extending from left to right and pointing to the beginning word of a sentence (Forgiveness is...crushed it.) in order to command that the sentence be indented. The sentence is thereafter indented a standard number of spaces, or may be indented to equal the length of

the arrow, as determined by user definition in the arrow logic that defines the green arrow. The recognition of this arrow logic may depend primarily on the context in which the arrow is drawn: an arrow extending from the left margin generally horizontally to the first word or words of a line; alternatively, this arrow transaction may be specified for a particular arrow, such as the green arrow shown. Such a green arrow logic could read: "indent the sentence that the arrow is pointing to by 8 spaces: or it could read: "indent the sentence that the arrow is pointing to by the distance that the arrow overlaps the sentence's text."

Other examples of format editing are shown in Figure 16. One or more words ("fragrance") may be selected by drawing a loop about it in blue line, and then writing the letter "I" with a line or arrow extending to one of the selected words. The software interprets the input as a command to italicize the selected words, which is carried out as shown in the Edited Text Result of Figure 16. Any other text effects known in the prior art may be accomplished in this manner, based on either an interpretation of letter commands such as "I", and/or based on user defined associations of letters (or numbers or symbols) with editing operations.

Alternatively, the user may draw a rectangle representing a switch, and then label the switch "*i*" for italic. If the switch is drawn in a color that has been used to select text portions, then the user may actuate the switch to cause all the text portions selected by that color to be converted to italic font. Other switches in other colors may be drawn to command the same or other format changes in text selected by such other colors. For instance, a red switch may be created, and

labeled with a "**B**" for bold, and pushing this switch will convert all text selected by red lines to bold font.

As another example, the words "heal" and "vessil" may be selected, and a red line drawn from at least one red selected text portion to the handwritten letters "sp", thereby commanding a spell check operation on all the words selected in the same manner as "heal" and "vessil"; that is, selected in this case with a red line. (Multiple words, sentences, and paragraphs may be selected and spell checked in this manner as well.) The corrected words "heel" and "vessel" are then substituted in the Edited Text Result.

In a further example of editing operations, words may be deleted merely by drawing a line through them, such as the words "mastery of fear," so that the sentence is restated without those words, as shown in the Edited Text Result. Here the transaction (removing the words) may be determined by the context of the lines drawn through the words, or by the color of the lines, as defined in an Info Window by a user.

Another editing maneuver that is difficult in the prior art and remarkably easy in this invention is the hand drawn GO TO command. As shown in Figure 10, a user may draw a star figure in a selected color (or any other figure and color combination; here, magenta) at any point in a text presentation onscreen. The interpretation of this magenta object (in combination with other attributes such as shape, location, orientation, and context) may be to function as a placeholder in the text. Thereafter, whenever a magenta star is drawn, and then an arrow (here in orange line) is drawn to the star, the software interprets these hand drawn

inputs as a command to Go To the place in the text where the magenta star was drawn. This simple technique, like all others described herein, obviates the need for pull down menus to gain access to a wide range of functionality. It may be appreciated that the star placeholder may be drawn in any one of a plurality of colors provided in the software palette, so that a large number of discrete placeholders may be drawn on the text, and accessed as shown in Figure 10. Thus the user may draw a green star or a teal circle and a Go To arrow extending thereto, and the display will immediately go to wherever these objects appear in the text.

A further editing operation available to the user is related to the use of voice recognition software for text input. As shown in Figure 15, a text input (whether from a voice recognition application or other input means) may be corrected by selecting a plurality of words in the text block using open or closed loops drawn in a selected color (here, blue), or any other selection technique described herein, as, for example, as shown in Figure 7A-7F. The user may draw a simple microphone symbol, as shown, and then draw an arrow from the selected words to the microphone symbol. The function assigned to the color of the arrow commands that the microphone generates a visible indication (such as blinking, changing color, etc.) to indicate that the microphone of a voice recognition system has been switched on and that the system is active. The user may then speak the correct words to replace the selected words, whereby corrections may be directed more simply than is now possible using voice recognition editing

commands alone. Tapping the microphone symbol may serve to turn the microphone ON and OFF.

The text editing software of the invention is also capable of carrying out outline and heading formatting. As shown in Figure 11A, the Raw Text contains a sentence and an enumeration of items that could be displayed more effectively as a list. The user may apply a list format to these items by drawing a short vertical line after each item to be placed in a list. The software interprets these hand drawn lines as commands to create a list, causing the items to be displayed in a list as shown in the Edited Text Result of Figure 11A. This transaction may be determined by the color of the lines drawn in Figure 11A, or by their context (i.e., vertical, placed between words, etc.).

With regard to Figure 11B, another technique for placing the enumerated items in a list format involves the user selecting the text portion with a loop terminating in an arrow that points to the word "List". The software applies the list command to the selected text, and the result is the listed items as shown in the Edited Text Result.

With regard to Figure 12, the list created in Figure 11 may be further formatted to become an outline. The list includes words such as " Produce", "Dairy", "Meat", and "Poultry", which may constitute general headings for the items appearing below each of these words. A line may be drawn vertically from the leftmost extent at the beginning of the sentence, then horizontally for a selected distance to set the indentation spacing, and thence vertically through the words of the list. The line is drawn in a manner to extend about the leftmost

portion of each word that is designated to be a heading, as shown in Figure 12. The hand drawn line is interpreted by the software as a command to create a numerical or alphabetical order (or the equivalent) of the heading words and indent them from the initial word of the sentence, and to number and further indent all of the words under each heading. All of this is accomplished by the user drawing one line onscreen (given that the line color and configuration are set up to be recognized for this purpose), resulting in the outline shown in the Edited Text Result of Figure 12. In contrast, any prior art word processing program requires many more input steps to achieve the same result.

The outline created as shown in Figure 12 is formally improper, due to the fact that some of the items listed are followed by commas, some by periods, and one has no punctuation at all. With regard to Figure 13A, one technique provided by the present invention for correcting the punctuation involves drawing a line that loops about all the punctuations that require correction, including the one instance (the word "Turkey") where no punctuation is present. Thereafter, an arrow (in the color that is defined to provide the desired transaction), is drawn from the selection line to a word (or symbol) representing the desired punctuation (here, Period). The desired effect is portrayed in the Edited Text Result of Figure 13A, in which every item of the outline is followed by a period.

As shown in Figure 13B, an additional feature of the outline technique enables the user to move items or sections in the outline while maintaining the proper numeration and sub heading and associated indentation. For example, if

the user desires to move the "B. Dairy" heading and dependent items, the user may draw a selection loop about the section and lead it to a blue check mark (for purposes of example only). This action assigns or copies this section of the outline to the blue check mark. Thereafter, the user may draw a blue check mark at some other portion of the outline (herein, at the end) to cause insertion of the selected outline portion at the point of the check mark. It is important to note that the software automatically renumbers (or re-alphabetizes) the sections properly in their new order, as shown by the Edited Text Result in Figure 13B.

With regard to Figure 14, the definition of the arrow logic that corrects the punctuation in Figure 13 is stated as item A: "The text that the arrow is drawn from will have its punctuation replaced by the punctuation definition that the arrow is pointing to." This definition associates the stated transaction with the blue arrow, and may be modified by the user to suit any particular purpose or to utilize any selected arrow color or style. Note: If a user wishes to change the color of an arrow assigned to this logic, he would select a new color and draw an arrow with that color directly on the arrow logic menu in such a way that the new colored arrow is closely adjacent or intersects the stated logic (the sentence that describes the arrow logic). Then this arrow logic text will be assigned to the new color arrow and the old color arrow's assignment to this arrow logic will be automatically deleted.

Returning to the outlining function, the line technique displayed in Figure 12 may be applied not only to a list, but also to sentences within a text block. As shown in Figure 17, a vertical line may be drawn through a plurality of sentences,

and each sentence that comprises an outline heading is selected by forming the line to extend in rectilinear fashion about the left end of the first word of the selected sentence. Alternatively, as also shown in Figure 18, the selected heading sentences may be selected by drawing the appropriate letter adjacent to the sentence and extending a line from the drawn letter to the beginning of the respective sentence. In either case, the outcome is the outline format shown in the Edited Text Result of Figures 17 and 18. Note that the problems associated with prior art heading systems, such as headings applied automatically to every paragraph, are eliminated.

A further formatting function of the present invention is depicted in Figures 19A-19C. A list of words may be acquired as a text block displayed onscreen, the text block titled "Instruments" and including a plurality of instrument names. A user may wish to apply the function contained in a folder object (here, a red folder that contains, for example only, an equalization function) to the instruments of the list, and this may be accomplished by drawing an orange arrow (which has a transaction defined as "process the elements that the arrow is drawn from with the process contained in the object that the arrow is drawn to.") from the Instrument list to the folder object. This arrow transaction, as shown in Figure 19A, applies to all the instruments in the list, due to the fact that the list is single text object on the display screen.

However, the user may wish to apply the function contained in the red folder to only one of the instruments listed. As shown in Figure 19B, the user may convert the text block titled "Instruments" from a single text object to a

plurality of onscreen objects, each object comprising one instrument. This may be accomplished by drawing an arrow extending to the text block. Here, the blue arrow has a transaction defined to be that the list is broken into individual objects, either word by word, or line by line, or the like. The result, as shown in Figure 19C, is that the instruments are converted to individual objects, as connoted by the highlighting in blue font (corresponding to the blue arrow). Thereafter, as shown in Figure 19C, the red folder function may be connected to a single instrument, such as the cellos as shown, by drawing an arrow from the red folder to the cello object onscreen, the red arrow defined as conveying the process contained in the red folder to the instrument to which the red arrow is pointing; i.e., the Cellos. No other instrument in the list will be subject to the equalization function embodied therein. (Multiple arrows could be drawn to other selected instruments in the list, as well.)

One real advantage of converting text to objects is that "text objects" themselves can represent things that are not merely text. In this case, "cellos" could represent a recording, i.e., a .WAV file, of cellos. So, drawing an arrow from the red EQ folder to the text object 'cellos" actually changes that equalization of the .WAV file called "cellos". This technique permits text to be used to edit audio, video, data files and any other type of computer file available on a computing system.

Any document or text block or onscreen object that is edited or processed as described above may be transmitted via email or internet to a recipient, either as the original text with all of the hand drawn editing inputs, or as the edited text

result. The original with the edit marks may enable the recipient to note the changes that have been made, and to alter those changes if desired and retransmit them to the sender.

With regard to Figure 20A, a further use for the present invention involves internet communications. A web browser usually provides a file titled Bookmarks (or equivalent) that lists web site names and corresponding web addresses. The user typically adds items to this list as new and interesting web sites are discovered. The invention provides that a user can draw a myriad of onscreen objects (here a green triangle) and then assign any number of bookmarks to this object. This object would then contain a series of websites (the websites that have been assigned to the object). In this example, the user may draw a blue selection line that selects a number of music websites listed as bookmarks. Then an arrow is drawn from the selection line to the green triangle. This line would need to have an appropriate logic assigned to it, like: "copy all text that the arrow is drawn from (the selected websites) and place it inside the object that the arrow or line is pointing to".

Thus an arrow from the selection line to the green triangle is interpreted by the software as a command to place all selected items within the green triangle. Thereafter, drawing a green triangle onscreen, as shown in Figure 20B, calls forth the list of music web sites contained therein. Thus the user may sort the Bookmarks in several distinct groups, according to interest and need.

The hand drawn text editing system may also be applied to email messages, as shown in Figure 21. An original email message as received may contain some

portions that the receiver wishes to save, or incorporate in another document, or forward to other individuals. However, it may also require editing before being put to such uses. In a typical email program, it is not possible to edit the received message; rather, the message text must be selected and pasted into a new message frame or into a word processing program document. As shown in Figure 21, the hand drawn editing techniques described above may be used to correct spelling errors, delete text, and the like, and the result is the Altered Email Message. It is noted that the altered email is in condition for retransmission. Also, the user may send the original email message including the hand drawn editing inputs, so that the further recipient may change the editing (e.g., for a working document developed by a group process) for further transmission.

Referring to Figure 22, another aspect of the invention enables the user designate one or more words to be graphic objects. As a graphic object, a word, phrase, sentence, etc., can be used to represent any number of definable items, such as, sound files, picture files, graphic files, video files, data files. Referring to Figure 22, each name shown in the list is a graphic object representing an entire investment portfolio pertaining to each name in the list. For instance, "Brown" represents an investment portfolio invested primarily in "Deutsche marks", whereas "Cohen" represents an investment portfolio invested primarily in Israeli shekels, and "Lemelson" represents an investment portfolio invested primarily in rubles. Furthermore, "James", "Dawson", and "Pines" represent investment portfolios primarily invested in US dollars. The Info Window for each named investor will display the individual investments for that investor.

The user may select a color and use this color to create a group of selected investors, by drawing a line (or loop or other selection technique described herein) to each name. As shown in Figure 22, a green line has been used to select "Brown", "Cohen", "Carroll", and "Lemelson", which exist as a single group distinguished by the color green. Likewise, a blue line has been used to select "James", "Dawson", and "Pines", which exist as a single group distinguished by the color blue. Thereafter, the user may draw a green switch and a blue switch, and label the green switch with a dollar symbol and the blue switch with a pound symbol. Then, when the green switch labeled dollar is touched, all of the investment portfolios in the green group will be converted to valuation in US dollars. Likewise, when the blue switch labeled pounds is touched, all the investment portfolios in the blue group will be converted to valuation in pounds.

This type of valuation conversion is common in financial transactions, and there are software applications known in the prior art to carry out such conversions. The system of the present invention is capable of directly causing the currency conversion by purely graphical means as shown in Figure 22..

The foregoing description of the preferred embodiment of the invention has been presented for purposes of illustration and description. It is not intended to be exhaustive or to limit the invention to the precise form disclosed, and many modifications and variations are possible in light of the above teaching without deviating from the spirit and the scope of the invention. The embodiment described is selected to best explain the principles of the invention and its practical application to thereby enable others skilled in the art to best utilize the

invention in various embodiments and with various modifications as suited to the particular purpose contemplated. It is intended that the scope of the invention be defined by the claims appended hereto.

5